数理解析研究所講究録
427巻 1981年 15-42

A GRAPH THEORETICAL APPROACH TO THE PERMUTATION LAYOUT*

S. Tsukiyama+, E. S. Kuh++, and I. Shirakawa+

\+ Department of Electronic Engineering, Faculty of Engineering, Osaka University, Suita, Osaka 565, Japan
++ Department of Electrical Engineering and Computer Sciences and Electronics Research Laboratory
University of California, Berkeley, CA 94720, U.S.A.

1. Introduction

The permutation layout problem introduced by Cutler-Schiloach[1] is not only of theoretical interest as the planar routing problem, but also has possibilities for practical applications to the wiring of PCB (Printed Circuit Board), to the layout for hybrid IC's, and to the routing of gate arrays (the master slice).

In Reference [1] three types of permutation layout are given, namely: packed-packed layout, packed-spaced layout, and spaced-spaced layout. Among them, only in the packed-packed layout, a good algorithm for realization has been proposed. The packed-spaced layout algorithm is similar to the routing method in [2,3]. However, no algorithm was proposed for the spaced-spaced layout, which is the most general case.

As pointed out by Shirakawa[4], the permutation layout

---

* This work was supported in part by the Grant in Aid for Scientific Research of the Ministry of Education, Science, and Culture of Japan under Grant: Cooperative Research (A) 435013 (1980).

problem can be transformed into a single-row single-layer routing problem. Thus the technique introduced in the single-row case can be used to solve the permutation layout problem. In this paper, we introduce a more sophisticated transformation and solve the spaced-spaced layout problem with the use of a graph theoretical algorithm.

2. Preliminary Definitions

2.1 Permutation Layout Problem

Let us consider two parallel horizontal lines called upper and lower rows, respectively, and consider nodes placed on these rows, as shown in Fig. 1(a). A net is a set of nodes to be connected by conductor lines which are composed of horizontal and vertical line segments. A net list is a set of disjoint nets. A realization of net list is a set of conductor lines, each of which connects all nodes in a net and does not intersect any other conductor lines.

Given a permutation

$$\pi = \begin{pmatrix} U \\ W \end{pmatrix} = \begin{pmatrix} N_1^u & N_2^u & \cdots & N_n^u \\ N_1^w & N_2^w & \cdots & N_n^w \end{pmatrix},$$

consider a net list $L^2(\pi) = [U,W]$ such that nodes u_i and w_i are contained in nets N_i^u and N_i^w, respectively. Namely, each net in $L^2(\pi)$ is composed of a pair of nodes; one is located in the upper row and another in the lower row.

For example, given a permutation $\pi = \begin{pmatrix} U \\ W \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 6 & 5 & 2 & 3 & 1 \end{pmatrix}$, the net list $L^2(\pi) = [U,W]$ and a realization of it are shown in Fig. 1(b).

In the realization of a net list $L^2(\pi)$, a crossing number X_u on the upper row is the number of intersections between the upper row and the conductor lines, not counting the intersections at nodes. A crossing number X_w on the lower row is defined similarly. For example, in a realization shown in Fig. 1(b), $X_u=2$ and $X_w=3$. A crossing number X is the maximum of X_u and X_w.

The Permutation Layout (Spaced-Spaced Layout) Problem (abbriviated PL) that we shall consider in this paper is stated as follows:

PL Problem: Given a permutation $\pi = \binom{U}{W}$, find a realization of net list $L^2(\pi)$ = [U,W] with the minimum crossing number X.

Now, we impose a restriction on the pattern of the conductor lines.

Restriction: We do not allow the conductor line for a net to run from the upper low to the lower row more than once.

The restriction is illustrated in Fig. 1(c), in which (I) is allowed, but not (II). With this assumption, we can see that there always exists a horizontal line between the upper and lower rows, which intersects exactly once with any conductor line of net as shown in Fig. 1(b). Let us call such horizontal line the middle row. Let M = (N_1^m, N_2^m, $\cdots$, N_n^m) be a sequence of nets on the middle row such that for $1 \leq i \leq n$, N_i^m indicates the net of the ith intersection.

We take the following approach to attack the problem PL:

Step I: Given a permutation $\pi = \begin{pmatrix} U \\ W \end{pmatrix}$, construct a sequence $M = (N_1^m, N_2^m, \cdots, N_n^m)$ of integers 1 through n. Then, create two permutations $\pi_u = \begin{pmatrix} U \\ M \end{pmatrix}$ and $\pi_w = \begin{pmatrix} M \\ W \end{pmatrix}$.

Step II: Find a realization of $L^2(\pi_u) = [U, W_u = M]$ such that the crossing number X_u on the upper row is minimal and in addition, there exists no conductor line below the middle row as shown in Fig. 2(a). Also, find a realization of $L^2(\pi_w) = [U_w = M, W]$ such that the crossing number X_w on the lower row is minimal and there exists no conductor line above the middle row as shown in Fig. 2(b).

For example, for the net list $L^2(\pi)$ given in Fig. 1(b), if sequence (6,4,1,2,3,5) is generated as M, and if realization of $L^2(\pi_u)$ and $L^2(\pi_w)$ are given as shown in Fig. 2(a) and (b), respectively, then the realization of $L^2(\pi)$ is obtained by combining these two realizations, which is the same as in Fig. 1(b).

The problem of finding a realization of $L^2(\pi_w)$ is similar to that of $L^2(\pi_u)$ since the problems are the same if we turn two rows of $L^2(\pi_w)$ upside down. Thus, our problem of PL is reduced to the following two problems:

Half-PL Problem: Given a net list $L^2(\pi_u) = [U, M]$, find a realization with a minimum crossing number X $(= X_u)$ such that no conductor line passes below the lower (i.e., middle) row.

Middle Sequence Problem (MSP): Given a permutation $\pi = \begin{pmatrix} U \\ W \end{pmatrix}$, find a sequence M which minimizes $\max[X_u, X_w]$, where X_u and X_w are the minimum crossing numbers of realizations of $L^2(\pi_u) = [U,M]$ and $L^2(\pi_w) = [M,W]$, respectively.

2.2 Interval Graphical Representation

In this section, we introduce the interval graphical representation[5] for formulating and solving the Half-PL problem.

Given a net list $L^2(\pi_u) = [U,M]$, let us consider two subsequences M_L and M_R of M with $M_L\_M_R = M$, where A_B represents a concatination of two sequences. Construct a sequence $M_L\_U\_M_R$ by concatinating M_L, U, and M_R in this order, and consider 2n nodes on the single row as single-row problem. For example, given a net list $L^2(\pi_u) = [U,M]$ and subsequences M_L and M_R shown in Fig. 3(a), those nodes on the single row are shown in Fig. 3(b). Let us denote the net list defined on these nodes by $L^1(M_L\_U\_M_R)$.

The interval graphical representation of the net list $L^1(M_L\_U\_M_R)$ on the single row is defined in the same way as [5]. For example, given a net list shown in Fig. 3(b), consider an ordering $f : L^1(S) \rightarrow \{1,2,\cdots,n\}$ such that $f(6)=1$, $f(5)=2$, $f(3)=3$, $f(4)=4$, $f(2)=5$, and $f(1)=6$, then the interval graphical representation associated with f is depicted as in Fig. 3(c), where each horizontal line corresponding to a net is arranged according to the ordering f from top down. Nodes which pertain to a net are marked as shown. Let us define the reference line[5] in an interval graphical representation: Introduce fictitious nodes

(0) and (∞) on the top-left and the top-right of the representation, respectively. Connect node (0) to the node u_1, which belongs to the first net in sequence U, with a line segment. Then connect the nodes u_2, u_3, ..., u_n, and (∞) in succession from left to right serially with line segments, as shown in Fig. 3(c). This continuous line from (0) to (∞) is called the reference line.

Now, let us stretch out the reference line and map it into the upper row. In the mean time, place the nodes m_i ($1 \leq i \leq n$) on the lower row as shown in Fig. 3(d). In such topological transformation, each net represented by an interval line is transformed into a path composed of horizontal and vertical line segments. This gives a realization of the problem Half-PL.

In order to ensure that, in the realization of Half-PL Problem, conductor lines do not go beyond the lower row, we require that the following two conditions be satisfied:

C_L: For nets N_i^m and N_j^m in M_L with $i<j$, there holds $f(N_i^m) < f(N_j^m)$.

C_R: For nets N_i^m and N_j^m in M_R with $i<j$, there holds $f(N_i^m) > f(N_j^m)$.

It is clear that for each interval graphical representation associated with an ordering satisfying conditions C_L and C_R, there corresponds a unique realization of Half-PL Problem. Furthermore, the crossing number of such a realization is simply the number of intersections between the reference line and the inteval lines, not counting the intersections at nodes. Let $X^f(M_L,U,M_R)$ be the number of such intersections between the reference line and the interval lines in the interval graphical representation of the net list $L^1(M_L\_U\_M_R)$ associated with ordering f. Then the crossing number X in the realization obtained by the above topological transformation is equal to $X^f(M_L,U,M_R)$.

Therefore, problem Half-PL is formulated as follows:

Half-PL Problem: Given a net list $L^2(\pi_u) = [U,M]$, find subsequences M_L and M_R and an ordering $f : L^1(M_L\_U\_M_R) \rightarrow \{1,2,\cdots,n\}$ such that

i) $M_L\_M_R = M$,

ii) $(=C_L)$ for nets N_i^m and N_j^m $(i < j)$ in M_L, there holds $f(N_i^m) < f(N_j^m)$,

iii) $(=C_R)$ for nets N_i^m and N_j^m $(i < j)$ in M_R, there holds $f(N_i^m) > f(N_j^m)$

iv) $X^f(M_L,U,M_R)$ is minimum.

Without loss of generality, we may assume that a net list $L^2(\pi_u) = [U,M]$ of the problem Half-PL does not have any net consisting of an isolated node or any net containing two consecutive nodes on a row.

Also, inherent in the approach of using the interval graphical representation, several patterns of conductor lines are excluded from considerations. These are shown in Fig. 4, where (a) indicates a forward-backward zigzagging around a row and (b) a combination of conductor lines which can not be generated by the method of the interval graphical representation.

3. Merging Algorithm

In this section, we propose a graph theoretical algorithm for problem Half-PL. We adopt the following approach to tackle the problem Half-PL.

<Algorithm for Half-PL>

Input : A net list $L^2(\pi_u) = [U,M]$.

Output: Subsequences M_L and M_R of M and an ordering $f : L^1(M_L \_U\_ M_R) \to \{1,2,\cdots,n\}$ such that f satisfies conditions C_L and C_R and $X^f(M_L,U,M_R)$ is minimum.

Step I: Let $M \triangleq (N_1^m,N_2^m,\cdots,N_n^m)$, then set $M_L \leftarrow (N_1^m)$ and $M_R \leftarrow (N_2^m,N_3^m,\cdots,N_n^m)$. Put $i \leftarrow 1$ and $X \leftarrow \infty$.

Step II: Solve the following problem called Simple Half-PL Problem.

Simple Half-PL Problem: Given a net list $L^1(S) = L^1(M_L \_U\_ M_R)$, find an ordering $f : L^1(S) \to \{1,2,\cdots,n\}$ such that f satisfies C_L and C_R, and $X^f(M_L,U,M_R)$ is minimum.

If for the solution f to Simple Half-PL, there holds $X^f(M_L,U,M_R) < X$, then store M_L, M_R, and f as the current solution to problem Half-PL, and set $X \leftarrow X^f(M_L,U,M_R)$.

Step III: Set $i \leftarrow i+1$. If $i \leq n-1$, then return to Step II by setting $M_L \leftarrow M_L \_ (N_i^m)$ and $M_R \leftarrow (N_{i+1}^m,N_{i+2}^m,\cdots,N_n^m)$; else terminate.

If Simple Half-PL is solved in polynomial time, then Half-PL is also solved. So, let us consider problem Simple Half-PL, in the following.

From conditions C_L and C_R, we can see that all possible orderings satisfying C_L and C_R correspond one-to-one to the sequences obtained by merging M_L with $\overline{M}_R$, where $\overline{M}_R \triangleq (N_n^m, N_{n-1}^m, \cdots, N_{\ell+1}^m)$ for $M_R = (N_{\ell+1}^m, N_{\ell+2}^m, \cdots, N_n^m)$. Namely, for merged sequence $Q \triangleq (q_1, q_2, \cdots, q_n)$ of M_L and $\overline{M}_R$, consider ordering f such that $f(q_i)=i$ $(i=1,2,\cdots,n)$, then this f automatically satisfies C_L and C_R. Conversely, for an ordering f satisfying C_L and C_R, consider the sequence $Q \triangleq (q_1, q_2, \cdots, q_n)$ such that $q_i = f^{-1}(i)$, then sequence Q is a sequnece obtained by merging M_L with $\overline{M}_R$.

From this observation, we can estimate the number of all possible ordering as follows:

$$\#\text{ of possible orderings} = \frac{|M_L+M_R|!}{|M_L|!\ |M_R|!} \sim \frac{n!}{(n/2)!\ (n/2)!}$$

$$\sim \frac{\sqrt{2\pi n}\ n^n\ e^{-n}}{[\ \sqrt{2\pi n/2}\,(n/2)^{n/2} e^{-n/2}]^2} = \frac{\sqrt{2}\ 2^n}{\sqrt{\pi n}}.$$

Thus, an exhaustive search algorithm cannot lead to a polynomial time algorithm for Simple Half-PL Problem.

Now, let us define a labeled <u>grid digraph</u> $G=[V,E]$, in which we will see that all merged sequences of M_L and $\overline{M}_R$ correspond one-to-one to the <u>directed paths</u> from <u>source</u> to <u>sink</u>.

Let $M_L \triangleq (N_1, N_2, \cdots, N_\ell)$ and $\overline{M}_R \triangleq (N_n, N_{n-1}, \cdots, N_{\ell+1})^*$.

Then, each vertex corresponds to a pair of integers, and vertex set V is defined as

$V \triangleq \{\ \langle i,j\rangle \mid 1 \leq i \leq \ell+1,\ 1 \leq j \leq n-\ell+1\ \}$.

In particular, vertices $\langle 1,1\rangle$ and $\langle \ell+1, n-\ell+1\rangle$ are designated

---

* We have dropped the superscript m, for convenience, since there is no confusion.

as source and sink, respectively. Edge set E consists of two disjoint sets E_L and E_R defined as

$E_L \triangleq \{ (\langle i,j\rangle,\langle i+1,j\rangle) \mid 1 \leqq i \leqq \ell,\ 1 \leqq j \leqq n-\ell+1 \}$, and

$E_R \triangleq \{ (\langle i,j\rangle,\langle i,j+1\rangle) \mid 1 \leqq i \leqq \ell+1,\ 1 \leqq j \leq n-\ell \}$.

Each edge $(\langle i,j\rangle,\langle i+1,j\rangle)$ in E_L has label net N_i, and each edge $(\langle i,j\rangle,\langle i,j+1\rangle)$ in E_R has label net N_{n-j+1}. Fig. 5 shows the grid digraph G for $M_L = (N_1,N_2,N_3)$ and $\overline{M}_R = (N_7,N_6,N_5,N_4)$.

For each directed path from source to sink, we can create a sequence of labels according to the edges passed by the directed path, which is a sequence obtained by merging M_L and $\overline{M}_R$. And we can easily verify that each directed path corresponds one-to-one to a merged sequence of M_L and $\overline{M}_R$. Therefore, the ordering f which satisfies both C_L and C_R corresponds one-to-one to the directed path from source to sink in the grid digraph G.

For example, the directed path shown by the bold-line in Fig. 5 corresponds to merged sequence $(N_1,N_7,N_6,N_2,N_5,N_3,N_4)$, and hence corresponds to ordering f : $f(N_1)=1$, $f(N_2)=4$, $f(N_3)=6$, $f(N_4)=7$, $f(N_5)=5$, $f(N_6)=3$, and $F(N_7)=2$.

Therefore, if we can assign an appropriate weight to each edge so that the total sum of the weights of all the edges on each directed path is exactly equal to the crossing number $X^f(M_L,U,M_R)$ in the interval graphical representation associated with the ordering f corresponding to the directed path, then we can solve problem Simple Half-PL by using a shortest path algorithm on the grid digraph. Note here that if weights assigned to edges satisfy the following two conditions, then the weights are appropriate ones.

(i) The weight of edge (<i,j>,<i+1,j>) with label N_i ($1 \leqq i \leqq \ell$) is equal to the number of intersections between the interval line of N_i and the reference line, which are caused if N_i is ordered between N_{n-j} and N_{n-j+1}. That is, N_i, N_{n-j}, and N_{n-j+1} are arranged in the order as $(\cdots, N_{n-j}, \cdots, N_i, \cdots, N_{n-j+1}, \cdots)$ in the merged sequence.

(ii) The weight of edge (<i,j>,<i,j+1>) with label N_{n-j+1} ($1 \leqq j \leqq n-\ell$) is equal to the number of intersections between the reference line and the interval line of N_{n-j+1}, which are caused if N_{n-j+1} is ordered between N_{i-1} and N_i.

Thus, let us consider how to assign such weights satisfying these conditions.

An interval between two consecutive nodes is called a unit interval, and the two nodes are designated as endnodes of the unit interval. A net containing an endnode of the unit interval is an end-net of the unit interval. If the net N_i^u containing the ith node u_i belongs to sequence M_L or M_R, then u_i is called an L-node or an R-node, respectively. A unit interval is called an L-L interval, an L-R interval, or an R-R interval, if both its endnodes are L-nodes, one is an L-node and another an R-node, or both are R-nodes, respectively, where the fictitious nodes (0) and (∞) are both regarded as L-nodes. The portion of the reference line for a unit interval H is denoted by RL(H).

Consider a unit interval H with end-nets N_a and N_b. As can be readily seen, only the interval lines of nets which cover interval H may or may not intersect RL(H), depending on the relative order with respect to N_a and N_b. Let us consider it in the following.

I. Let N_i be a net in M_L which covers H. (See Fig. 6)

i) H is an L-L interval. We can assume $a<b$ without loss of generality. In this case, RL(H) and N_i intersect each other when $a < i < b$. Thus, edges with label $N_i (a<i<b)$ must have weights corresponds to this intersection

ii) L-R interval. Let $N_b \triangleq N_{n-j+1}$

<u>Case 1 $(a<i)$.</u> In this case, RL(H) and N_i intersects, if and only if the ordering f satisfies $f(N_i)<f(N_b)$. Therefore, only edges $(<i,h>,<i+1,h>)$ with $h \leqq j$ must have weights corresponding to this intersection.

<u>Case 2 $(a>i)$</u>. In this case, RL(H) and N_i intersect, if and only if the ordering f satisfies $f(N_i)>f(N_b)$. Therefore, only edges $(<i,h>,<i+1,h>)$ with $h>j$ must have weights corresponding to this intersection.

iii) R-R interval. Let $N_a \triangleq N_{n-k+1}$ and $N_b \triangleq N_{n-j+1}$ and assume $k<j$ without loss of generality. In this case, RL(H) and N_i intersect, if and only if the ordering satisfies $f(N_a)<f(N_i)<f(N_b)$. Therefore, only edges $(<i,h>,<i+1,h>)$ with $k < h \leqq j$ must have weights corresponding to this intersection.

II. A similar analysis can be given for a net in $\overline{M}_R$.

Based on the above discussion, we can devise an algorithm of finding the desired weights for edges, by processing unit intervals successively from the left to the right. This is given in Appendix.

If we apply <Weight Assignment Algorithm> in Appendix to a net list $L^1(S)=L^1(M_L\_U\_M_R)$ shown in Fig. 7(a), then we have the weights for all edges shown in Fig. 7(b). In the Figure, the number in a bracket and the sequence of alphabets beside an edge show the weight of the edge and the unit intervals at which the weight of the edge is increased by one, respectively. The interval graphical representation shown in Fig. 7(a) is associated with the ordering corresponding to the directed path from source to sink through the lower left corner of the grid digraph. Hence, as shown by the weight and sequence (c,d) beside edge label N_1 on the directed path in Fig. 7(b), net N_1 has 2 intersections at unit intervals c and d in the interval graphical representation of Fig. 7(a).

Let the <u>length</u> of a directed path in the grid digraph be the sum of the weights of all edges on the directed path, then the following lemma can be readily verified from the above discussion.

<u>Lemma 1:</u> The length of a directed path in the grid digraph for a net list $L^1(S)=L^1(M_L\_U\_M_R)$ is equal to the crossing number $X^f(M_L,U,M_R)$ in the interval graphical representation associated with the ordering f corresponding to the directed path.

Thus, an optimum ordering f for Simple Half-PL is obtained by a shortest path algorithm on the grid digraph. In the following, we describe an algorithm for Simple Half-PL Problem.

<Merging Algorithm>

Input : A net list $L^1(S) = L^1(M_L \cup M_R)$.

Output: An ordering $f : L^1(S) \to \{1,2,\cdots,n\}$ such that f satisfies C_L and C_R and $X^f(M_L,U,M_R)$ is minimum.

Step 1. Create grid digraph $G=[V,E]$ for $M_L \triangleq (N_1^m, N_2^m, \ldots, N_\ell^m)$ and $\overline{M}_R \triangleq (N_n^m, N_{n-1}^m, \ldots\ldots N_{\ell+1}^m)$.

Step 2. <Weight Assignment Algorithm>.

Step 3. Compute shortest distance from source to each vertex $<i,j>$. Noting that in the grid digraph any directed path from source to vertix $<i,j>$ passes though vertex either $<i-1,j>$ or $<i,j-1>$, we can implement this process in the processing time propotional to the number of vertices of the grid digraph.

Step 4. Find a shortest path from source $<1,1>$ to sink $<\ell+1,n-\ell+1>$ by tracing back from sink to source.

By substituting <Merging Algorithm> for Step II in <Algorithm for Half-PL>, we can complete the algorithm, for which we have the following theorem.

Theorem 1: <Algorithm for Half-PL> can find an optimum solution to problem Half-PL in the processing time of order $0(n^4)$ and in the memory space of order $0(n^2)$.

Proof. We can easily see from Lemma 1 that the algorithm can find an optimum solution. Noting that $0(n^2)$ space is required for the grid digraph and that $0(n)$ space is sufficient for a given net list and for other sets and sequences, we can also verify that the algorithm is implemented in $O(n^2)$ space.

Let us consider the processing time. In <Merging Algorithm> Steps 1 and 3 can be executed in $O(n^2)$ time and Step 4 in $O(n)$ time. As is shown in Appendix, the total time required for <Weight Assignment Algorithm> is $O(n^3)$, and therefore the total time of <Merging Algorithm> is $O(n^3)$. Hence, the theorem has been proven, since the loop of Steps II-III in <Algorithm for Half-PL> requires n-1 times iterations. $\square$

4. Middle Sequence Problem

In this section, we consider the problem of finding a sequence M, for which a good algorithm has not been devised. However, we can construct a heuristic algorithm with a better upper bound for the crossing number than [1].

Let u_j and w_j $(1 \leq j \leq n)$ be nodes on the upper and the lower rows, respectively. For a net N_i, the subscript-number of node in N_i on the upper row is denoted by $u(N_i)$, and that on the lower row by $w(N_i)$. In the upper row, if the node $u_{u(N_i)}$ of net N_i is contained in the left-half of the upper row (i.e., $1 \leq u(N_i) \leq \lfloor n/2 \rfloor$), then net N_i is said to be an <u>upper-left net</u>, where $\lfloor x \rfloor$ denotes the largest integer not greater than x. Otherwise, N_i is called an <u>upper-right net</u>. Similarly, we define a <u>lower-left net</u> and a <u>lower-right net</u> on the lower row.

A set of nets can be partitioned into four subsets $\mathcal{S}_1$, $\mathcal{S}_2$,

$\mathcal{S}_3$, and $\mathcal{S}_4$ as follows.

$\mathcal{S}_1 \triangleq \{ N_i \mid N_i$ is a lower-left, upper-right net $\}$.

$\mathcal{S}_2 \triangleq \{ N_i \mid N_i$ is a lower-left, upper-left net $\}$.

$\mathcal{S}_3 \triangleq \{ N_i \mid N_i$ is a lower-right, upper-left net $\}$.

$\mathcal{S}_4 \triangleq \{ N_i \mid N_i$ is a lower-right, upper-right net $\}$.

In a sequence M, the nets are arranged in the order of $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, and $\mathcal{S}_4$ from the left to the right, as shown in the following algorithm.

<u><Middle Sequence Algorithm></u>

<u>Input</u> : A net list $L^2(\pi) = [U,W]$.

<u>Output</u>: A sequence M.

<u>Step 1</u>: Compute subsets $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, and $\mathcal{S}_4$ of nets.

<u>Step 2</u>: Let U1 and U2 be the left half subsequence $(N^u_1, N^u_2, \cdots, N^u_{\lfloor n/2 \rfloor})$ of U and the right half subsequence $(N^u_{\lfloor n/2 \rfloor+1}, N^u_{\lfloor n/2 \rfloor+2}, \cdots, N^u_n)$ of U, respectively. And let us define subsequences W1 and W2 of W similar to U1 and U2, respectively. Then, set $S1 \leftarrow \overline{W1} \cap \mathcal{S}_1$, $S2 \leftarrow \overline{U1} \cap \mathcal{S}_2$, $S3 \leftarrow U1 \cap \mathcal{S}_3$, and $S4 \leftarrow \overline{W2} \cap \mathcal{S}_4$, where for a set $\mathcal{S}$ and a sequence A, $A \cap \mathcal{S}$ represents a subsequence of A by deleting all the elements not contained in $\mathcal{S}$ from A without changing the order of nets in A, and $\overline{A}$ denotes a sequence with the reverse order of A.

<u>Step 3</u>: Set $M \leftarrow S1 \_ S2 \_ S3 \_ S4$.

An illustrative example of this algorithm is shown in Fig. 8.

As discussed in [1], let us consider the upper bound for the crossing number of a realization obtained by the proposed algorithm. We show that our algorithm gives a slightly better upper bound than [1] in what follows.

Given an interval graphical representation associated with ordering f, let us define an unconnected node and block similar to an unconnected point and block defined in [1]. For a net N_a in the representation, a node contained in a net N_b with $f(N_a)<f(N_b)$ is said to be an unconnected node for N_a, and a maximal set of consecutive unconnected nodes is called a block for N_a. Moreover, let α_i be the number of blocks for net $f^{-1}(i)$ and let β_i be the number of intersections between the reference line and the interval line of net $f^{-1}(i)$. Then, as seen from Fig. 9(a), we have the following lemma.

Lemma 2[1]: For a net $f^{-1}(i)$, there hold

$$\beta_i \leqq 2\alpha_i - 1 \text{ and}$$
$$\alpha_i \leqq \min[\, n-i+1,\ i\,].$$

Consider the interval graphical representations for $L^2(\pi_u) = [U,M]$ and $L^2(\pi_w) = [W,M]$ shown in Fig. 9(b). Let X_u^* and X_w^* be the crossing numbers of these representations, then we can easily verify from <Merging Algorithm> that $X_u \leqq X_u^*$ and $X_w \leqq X_w^*$, where X_u and X_w are the crossing numbers on the upper and the lower rows obtained by <Algorithm for Half-PL>, respectively. By using X_u^* and X_w^* and Lemma 2, we can show the following theorem.

Theorem 2: Let $L^2(\pi) = [U,W]$ be a given net list with n nets. Then, for the crossing number $X_1 \triangleq \max[\, X_u, X_w\,]$ obtained by the proposed algorithm, we have

$$X_1 \leqq n^2/8 + O(n).$$

Proof: Let us count X_u^* and X_w^*.

$$X_u^* = \sum_{i=1}^{n} \beta_i \leqq 2\left(\sum_{i=1}^{\lceil n/2 \rceil} \alpha_i + \sum_{i=\lceil n/2 \rceil+1}^{n} \alpha_i \right) - n$$

$$\leqq 2\left(\sum_{i=1}^{\lceil n/2 \rceil} \min[\, \lceil n/2 \rceil - i + 1,\ i \,] + |\mathcal{S}_2| \right) - n$$

$$\leqq 4\left(1 + 2 + \cdots + \frac{\lceil n/2 \rceil + 1}{2} \right) + 2|\mathcal{S}_2| - n$$

$$\leqq \frac{(n+5)(n+9)}{8} + 2|\mathcal{S}_2| - n$$

$$< \frac{(n+5)(n+9)}{8},$$

since $|\mathcal{S}_2| < \lfloor n/2 \rfloor$, where $\lceil x \rceil$ deontes the smallest integer not less than x.

$$X_w^* = \sum_{i=1}^{n} \beta_i \leqq 2\left(\sum_{i=1}^{\lceil n/2 \rceil} \alpha_i + \sum_{i=\lceil n/2 \rceil+1}^{n} \alpha_i \right) - n$$

$$\leqq 2\Bigg(|\mathcal{S}_4| + \sum_{i=|\mathcal{S}_4|+1}^{\lceil n/2 \rceil} \min[\, \lceil n/2 \rceil - i + 1,\ i \,]$$

$$+ |\mathcal{S}_1| + \sum_{i=|\mathcal{S}_1|+1}^{\lceil n/2 \rceil} \min[\, \lceil n/2 \rceil - i + 1,\ i \,] \Bigg) - n$$

$$\leqq 2\left(\sum_{i=|\mathcal{S}_4|+1}^{\lceil n/2 \rceil} \min[\lceil \tfrac{n}{2} \rceil - i + 1, i] + \sum_{i=|\mathcal{S}_1|+1}^{\lceil n/2 \rceil} \min[\lceil \tfrac{n}{2} \rceil - i + 1, i] \right)$$

$$+ 2\left(|\mathcal{S}_1| + |\mathcal{S}_4| \right) - n.$$

Noting that $|\mathcal{S}_1| + |\mathcal{S}_4| = \lceil n/2 \rceil$, there holds

$$X_w^* \leqq 2\left(\sum_{i=1}^{\lceil n/2 \rceil} \min[\, \lceil n/2 \rceil - i + 1,\ i \,] \right) + 2\lceil n/2 \rceil - n$$

$$\leqq \frac{(n+5)(n+9)}{8} + 1.$$

Thus, we have

$$X_1 \triangleq \max[\, X_u,\ X_w \,] \leqq \max[\, X_u^*,\ X_w^* \,] \leqq \frac{(n+5)(n+9)}{8} + 1$$
$$= n^2/8 + O(n). \qquad \square$$

In [1], no algorithm is proposed for a spaced-spaced layout. Therefore, their upper bound for a spaced-spaced layout is $n^2/4 + O(n)$, which is the same for a packed-spaced layout. We can see from the above discussion that the upper bound for a spaced-spaced layout is almost half of that for a packed-spaced layout, which can be expected from the definitions of both layouts.

Finally, it is easy to see that this <Middle Sequence Algorithm> is implemented in $O(n)$ time and $O(n)$ space.

5. Conclusion

In this paper, we have considered the permutation layout problem with the crossing number as a criterion for minimization, which is called the spaced-spaced layout problem in Ref. [1]. Our approach that we have taken to tackle this problem is to break up the original problem into two manageable subproblems; Half-PL Problem and Middle Sequence Problem. For the first problem, we have devised a polynomial time algorithm with the use of a shortest path algorithm on the weighted grid digraph, and for the second problem, we have proposed a heuristic algorithm with an analysis on the upper bound for the crossing number of the solution given by the algorithm.

As pointed out in Ref. [1], there still remain a number of intriguing problems regarding the permutation layout. Among them, of practically importance is the problem with the maximum number of conductor lines between two consecutive nodes on a row (the between-nodes congestion) as a criterion for minimization[6].

REFERENCES

[1] M. Cutler and Y. Shiloach, "Permutation layout," Networks, vol. 8, pp. 253-278, 1978.

[2] H. C. So, "Some theoretical results on the routing of multilayer printed wiring boards," Proc. IEEE Int. Symp. on Circuit and Systems, pp. 296-303, 1974.

[3] B. S. Ting, E. S. Kuh, and I.Shirakawa, "The multilayer routing problem: Algorithms and necessary and sufficient conditions for the single-row single-layer case," IEEE Trans. on Circuits and systems, vol. CAS-23, pp. 768-778, 1976.

[4] E. S. Kuh, T. Kashiwabara, and T. Fujisawa, "On optimum single-row routing," IEEE Trans. on Circuits and Systems, vol. CAS-26, pp. 361-368, 1979.

[5] I. Shirakawa, "Some comments on permutation layout," Networks, vol. 10, pp. 179-182, 1980.

[6] T. Ozawa and H. Sekiguchi, "A mapping of a planar graph on a plane and its application," Monograph CAS80-48, IECE Japan, pp. 33-40, 1980 (in Japanese).

Appendix: (Weight Assignment Algorithm)

<Weight Assignment Algorithm>

Input : A net list $L^1(M_L \cup M_R)$ and a grid digraph $G = [V,E]$ for $M_L \triangleq (N_1, N_2, \cdots, N_\ell)$ and $\bar{M}_R \triangleq (N_n, N_{n-1}, \cdots, N_{\ell+1})$.

Output: Weight $WT(e)$ for each edge $e \in E$.

Step 1: Set $WT(e) \leftarrow 0$ for every edge $e \in E$.

Step 2: Pick up the leftmost unit interval H, on which the following steps have not been conducted. If there is no such H, then terminate; else execute the followings on H.

Step 3: If H is an L-L interval, then go to Step 4. If H is an R-R interval, then go to Step 5. Otherwise, go to Step 6.

Step 4: Let N_a and N_b be end-nets of H with $a < b$.

1°. For each net N_i in M_L which passes through H and satisfies $a < i < b$, conduct (i).

(i). For every $e = (\langle i,h\rangle, \langle i+1,h\rangle)$ with $1 \le h \le n-\ell+1$,

$$WT(e) \leftarrow WT(e) + 1.$$

2°. For each net N_{n-j+1} in $\bar{M}_R$ which passes through H, conduct (ii).

(ii). For every $e = (\langle h,j\rangle, \langle h,j+1\rangle)$ with $a < h \le b$,

$$WT(e) \leftarrow WT(e) + 1.$$

3°. Then, return to Step 2.

Step 5: Let N_{n-a+1} and N_{n-b+1} be end-nets of H with $a < b$.

1°. For each net N_i in M_L which passes through H, conduct (iii).

(iii). For every $e = (\langle i,h\rangle, \langle i+1,h\rangle)$ with $a < h \le b$,

$$WT(e) \leftarrow WT(e) + 1.$$

2°. For each net N_{n-j+1} in $\overline{M}_R$ which passes through H and satisfies $a < j < b$, conduct (iv).

(iv). For every $e = (\langle h,j\rangle, \langle h,j+1\rangle)$ with $1 \le h \le \ell + 1$,

$$WT(e) \leftarrow WT(e) + 1.$$

3°. Then, return to Step 2.

Step 6: Let $N_a \in M_L$ and $N_{n-b+1} \in \overline{M}_R$ be end-nets of H.

1°. For each net N_i in M_L which passes through H and satisfies $a < i$, conduct (v).

(v). For every $e = (\langle i,h\rangle, \langle i+1,h\rangle)$ with $h \le b$,

$$WT(e) \leftarrow WT(e) + 1.$$

2°. For each net N_i in M_L which passes through H and satisfies $a > i$, conduct (vi).

(vi). For every $e = (\langle i,h\rangle, \langle i+1,h\rangle)$ with $h > b$,

$$WT(e) \leftarrow WT(e) + 1.$$

3°. For each net N_{n-j+1} in $\overline{M}_R$ which passes through H and satisfies $b < j$, conduct (vii).

(vii). For every $e = (\langle h,j\rangle, \langle h,j+1\rangle)$ with $h \le a$,

$$WT(e) \leftarrow WT(e) + 1.$$

4°. For each net N_{n-j+1} in $\overline{M}_R$ which passes through H and satisfies $b > j$, conduct (viii).

(viii). For every $e = (\langle h,j\rangle, \langle h,j+1\rangle)$ with $h > a$,

$$WT(e) \leftarrow WT(e) + 1.$$

5°. Then, return to Step 2.

Let us consider the processing time required in this algorithm. We can easily see that Steps 4, 5 and 6 are implemented at most in $O(n^2)$. Therefore, the total time required by the loop through Steps 2 to 6 is $O(n^3)$, and hence the total time required for <Weight Assignment Algorithm> is $O(n^3)$, since Step 1 is implemented in $O(n^2)$ time.

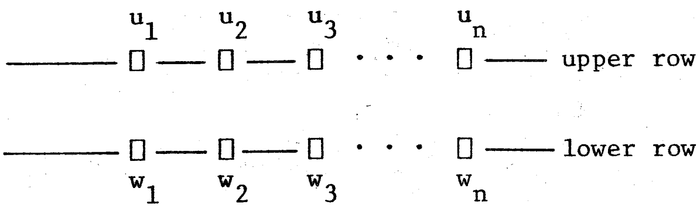


Fig. 1(a). Nodes on the upper and lower rows.

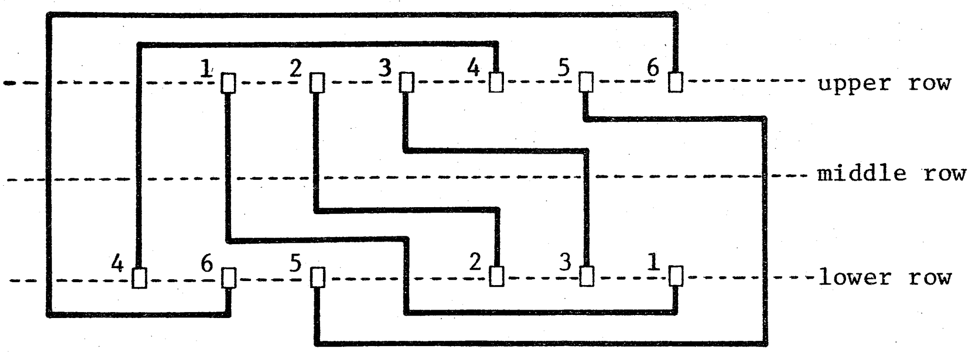


Fig. 1(b). A realization of net list $L^2(\pi)$ and the middle row.

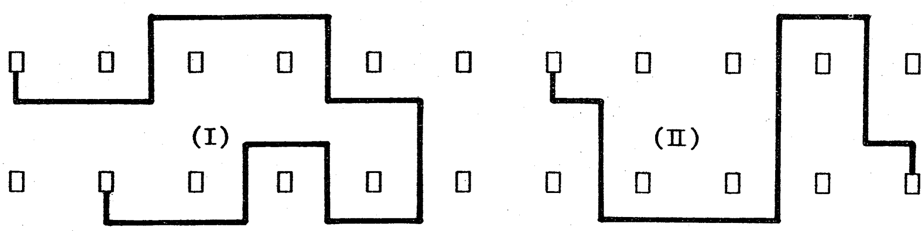


Fig. 1(c). Allowed pattern (I) and prohibited pattern (II).

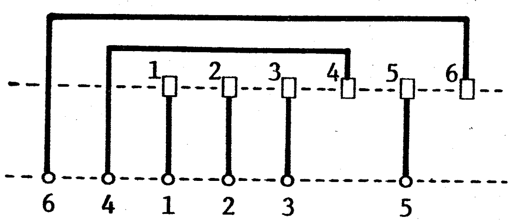


Fig. 2(a). A realization of $L^2(\pi_u)$.

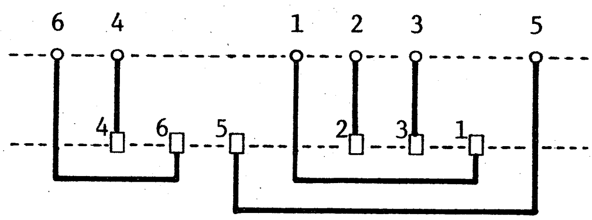


Fig. 2(b). A realization of $L^2(\pi_w)$.

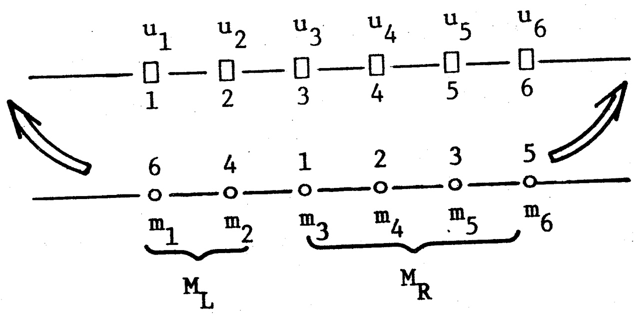


Fig. 3(a). Net list $L^2(\pi_u) = [U,M]$ and subsequences M_L and M_R of M.

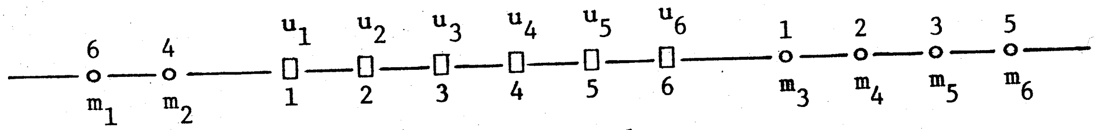


Fig. 3(b). Net list $L^1(M_L\_U\_M_R)$.

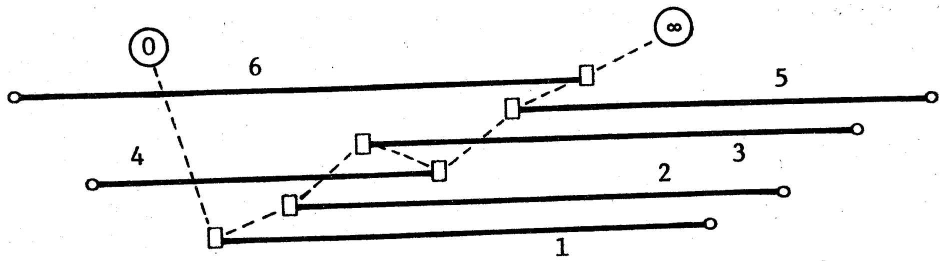


Fig. 3(c). An interval graphical representation of $L^1(M_L\_U\_M_R)$ and the reference line.

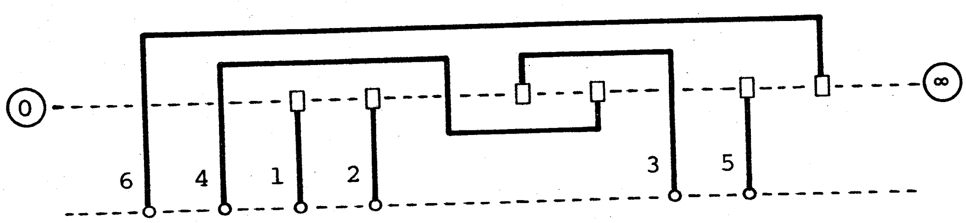


Fig. 3(d). A realization transformed from the interval graphical representation of (c).

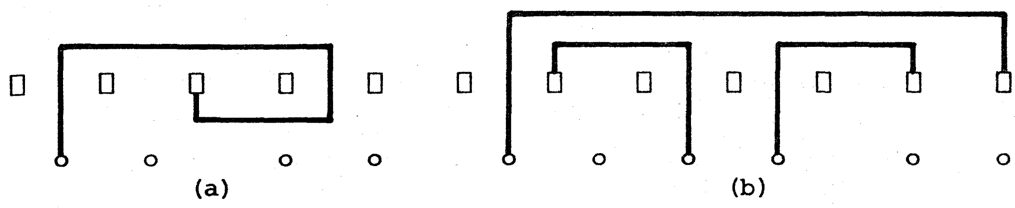


Fig. 4. Exceptions of routing patterns.

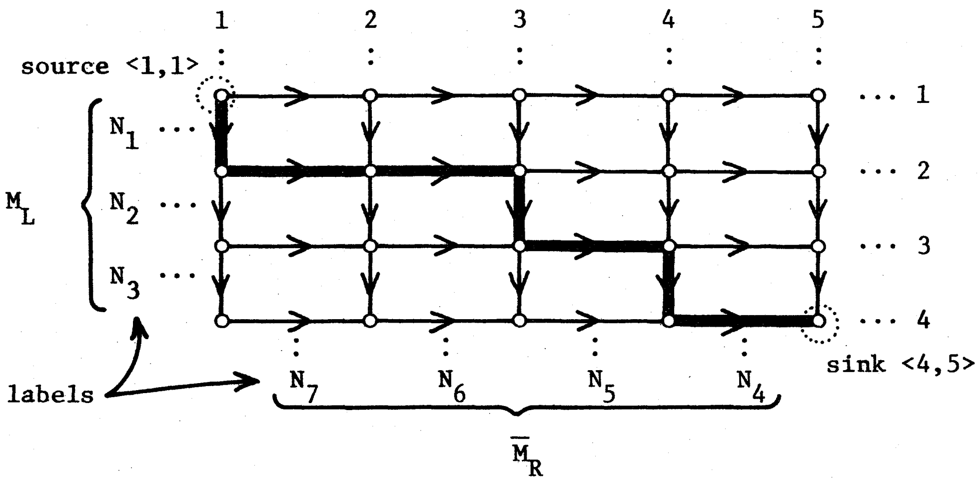
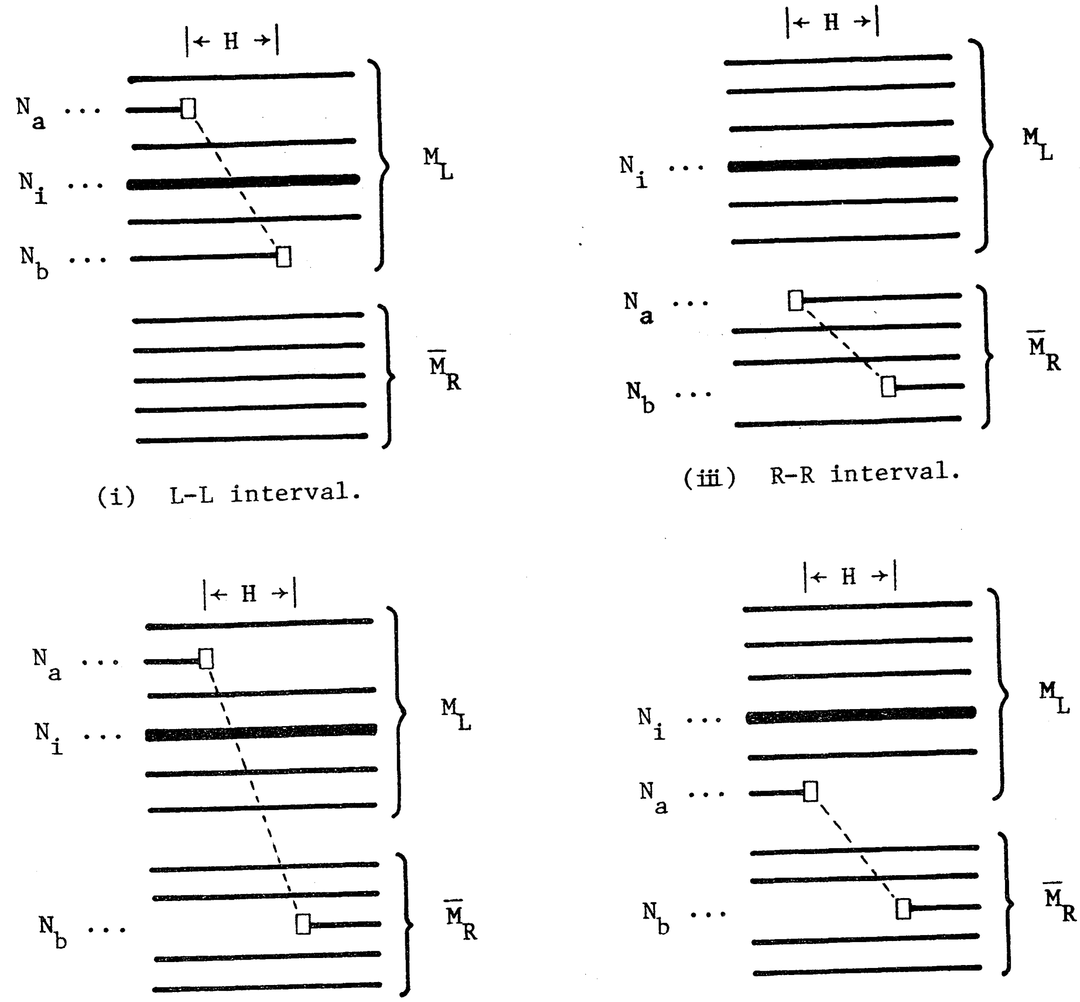


Fig. 5. Grid digraph G for M_L and $\overline{M}_R$, and a directed path corresponding to merged sequence $(N_1, N_7, N_6, N_2, N_5, N_3, N_4)$.

40



(i) L-L interval.

(iii) R-R interval.

(ii) L-R interval, Case 1.

(ii) L-R interval, Case 2.

Fig. 6. Unit interval H and net N_i which may intersect each other.

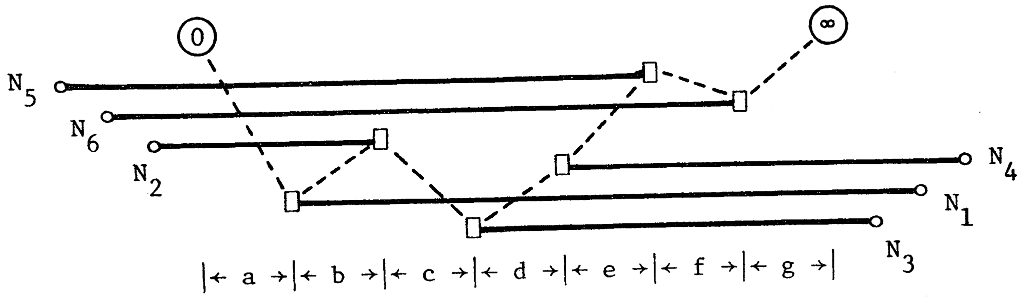


Fig. 7(a). Net list $L^1(M_L \_ U \_ M_R)$ and an interval graphical representation.

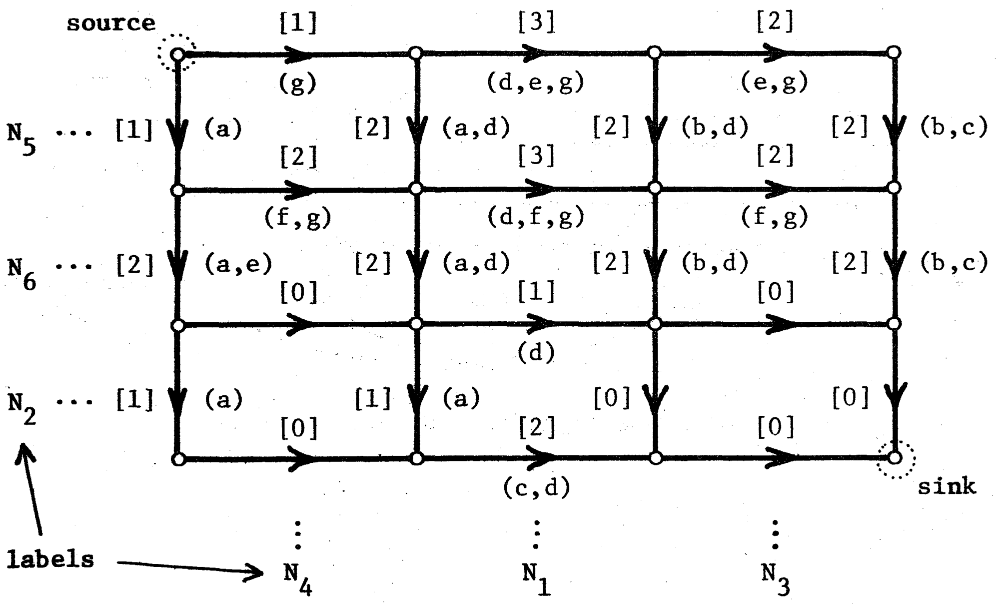


Fig. 7(b). Grid digraph for net list in (a) and weights of edges.

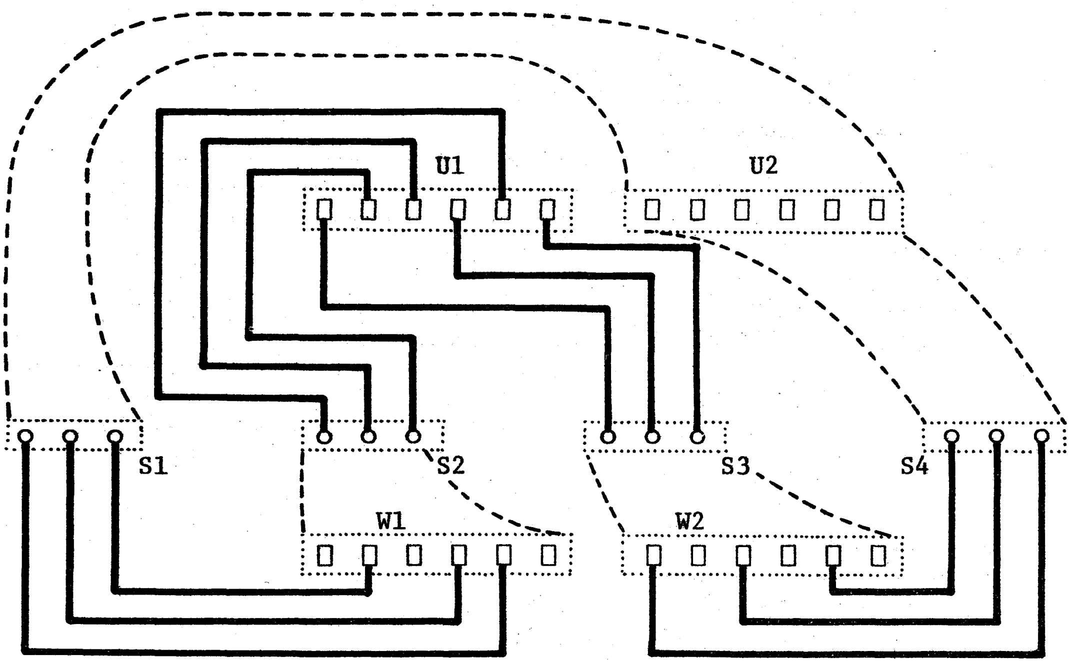


Fig. 8. A frame of middle sequence M.

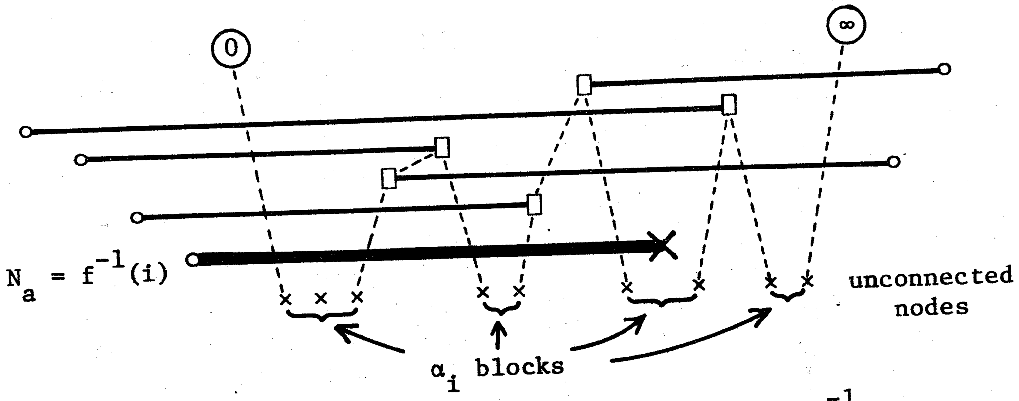


Fig. 9(a). Unconnected nodes and blocks for net $f^{-1}(i)$.

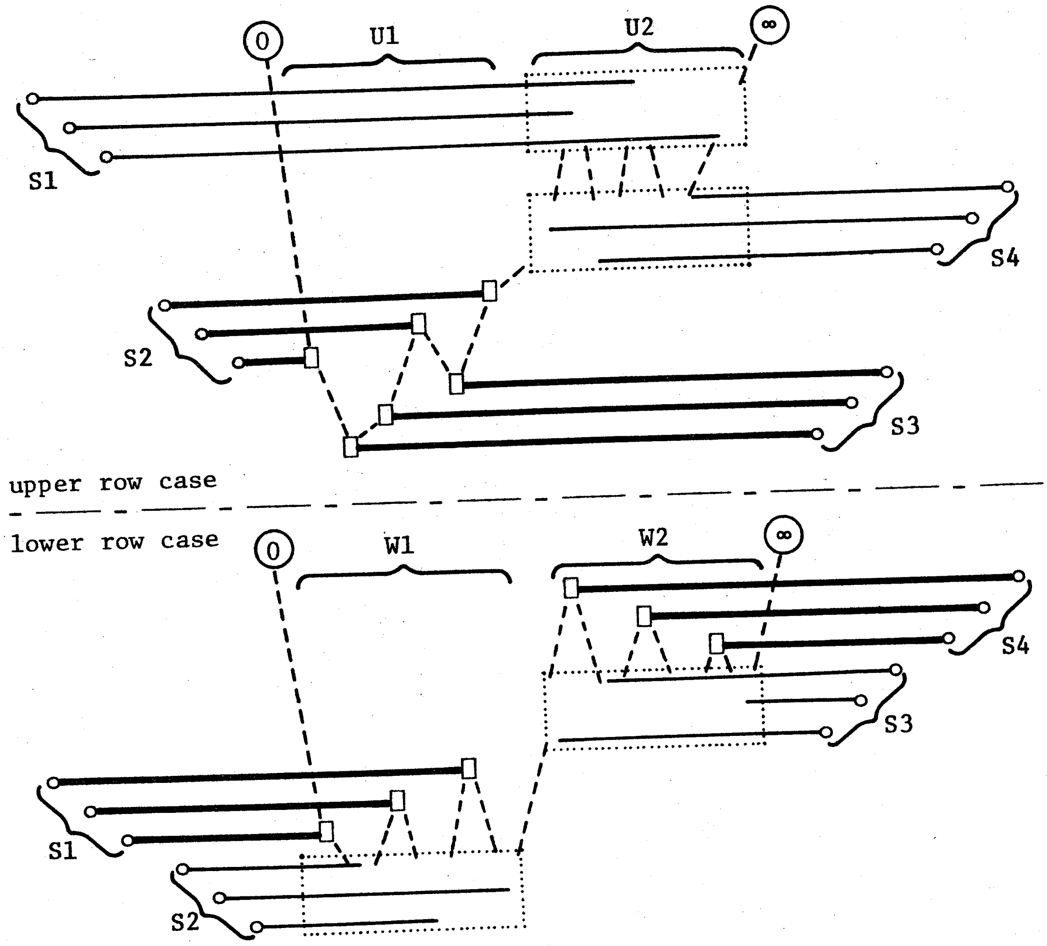


Fig. 9(b). Interval graphical representations of $L^2(\pi_u)$ and $L^2(\pi_w)$.